



(19) BUNDESREPUBLIK  
DEUTSCHLAND



DEUTSCHES  
PATENT- UND  
MARKENAMT

(12) Offenlegungsschrift  
(10) DE 199 14 454 A 1

(51) Int. Cl.<sup>7</sup>:  
**G 06 F 17/30**

(71) Anmelder:  
Dinglreiter, Marcus, 96224 Burgkunstadt, DE  
  
(74) Vertreter:  
BOEHMERT & BOEHMERT, 80801 München

(72) Erfinder:  
gleich Anmelder

**Die folgenden Angaben sind den vom Anmelder eingereichten Unterlagen entnommen**

(54) Rechnergestütztes System und Verfahren zum Aufbauen einer Datenbank

(57) Die Erfindung betrifft ein rechnergestütztes System zum Aufbauen einer Datenbank, umfassend wenigstens zwei Grundtabellen (tb1Key, tb1Mmo) für zu jeweils einer Grundklasse gehörende Elemente, wenigstens eine Beziehungstabelle (tb1KeyKey), die mit den Grundtabellen verbindbar ist, um Beziehungen zwischen den Elementen herzustellen, und eine Kategorietabelle (tb1ZCat), die mit der Beziehungstabelle verbindbar ist, um den jeweiligen Beziehungen eine eindeutige Beziehungskategorie zuzuordnen. Die Erfindung betrifft ferner ein Verfahren zum Aufbauen einer Datenbank in einer elektronischen Datenverarbeitungsanlage, mit folgenden Verfahrensschritten: Eingeben und Speichern von zu jeweils einer Grundklasse gehörenden Elementen in wenigstens zwei Grundtabellen (tb1Key, tb1Mmo), Eingeben und Speichern von Beziehungen, die zwischen Elementen in den Grundtabellen bestehen, in wenigstens einer Beziehungstabelle (tb1KeyKey, tb1KeyMmo) und Verbinden der Beziehungstabelle (tb1KeyKey, tb1KeyMmo) mit der Grundtabelle (tb1Key, tb1Mmo) und Eingeben und Speichern von Beziehungs-kategorien in einer Kategorietabelle (tb1ZCat) und Verbin-den der Kategorietabelle mit der Beziehungstabelle, um den jeweiligen Beziehungen eine eindeutige Beziehungs-kategorie zuzuordnen.

**DE 199 14 454 A 1**

**DE 199 14 454 A 1**

## Beschreibung

Die Erfindung betrifft ein rechnergestütztes System und ein Verfahren zum Aufbauen einer Datenbank.

Es ist allgemein anerkannt, daß die Verwaltung von Daten in einer Datenbank enorme technische Vorteile gegenüber einer "verstreuten" Datenhaltung hat. In einer Datenbank werden mehrfach vorkommende, redundante Daten vorzugsweise so organisiert, daß Datensätze nur ein einziges Mal angelegt werden müssen, was große Vorteile für die Datenspeicherung, jedoch auch die Verwaltung und Wartung sowie für die Datensicherung mit sich bringt. Datenbanken erlauben es, einzelne Anwendungen von den verwalteten Datenbeständen zu trennen, wobei mehrere, verschiedene Anwendungen auf einen gemeinsamen Datenbestand zugreifen können. Diese sowie weitere Vorteile von Datenbanken sind bekannt.

Grundsätzlich muß unterschieden werden zwischen der eigentlichen Datenbank und dem Datenbankverwaltungssystem oder dem System zum Aufbauen einer Datenbank. Das Datenbankverwaltungssystem stellt ein Entwicklungswerkzeug zur Erstellung, Kontrolle und Veränderung von Datenbanken dar. Die eigentliche Datenbank muß demzufolge erst mit Hilfe des Datenbankverwaltungssystems aufgebaut werden. Bei einer Datenbank wird weiter unterschieden, ob sie die zu verwaltenden Informationen selbst oder nur einen Verweis (Zeiger) auf die Informationen enthält, wobei Kombinationen hieraus denkbar sind. So kann z. B. bestimmte strukturierbare Information zur Erfassung (z. B. bei einer Literaturverwaltung Angaben zum Autor, Titel, Signatur etc.) in einer Datenbank zusammen mit Verweisen auf Dateien (z. B. Word- oder HTML-Dokumente) verwaltet werden, welche die eigentlich gesuchte Information (z. B. Publikationen) enthalten.

Traditionell gibt es verschiedene Arten von Datenbanken, wobei man die unter anderem Kategorien hierarchische, relationale, und objektorientierte Datenbanken unterscheidet.

In einer hierarchischen Datenbank werden die Daten entsprechend einer hierarchischen Gliederung in einer einzelnen sequentiellen Datei abgelegt. In einem Adressverwaltungssystem z. B. kann die höchste Hierarchiestufe der Name sein, also etwa 1 = Müller, 2 = Huber, 3 = Meier; die nächste Hierarchiestufe ist der Vorname, also etwa 1.1 = Hans (Müller), 2.1 = Peter (Huber), 3.1 = Fritz (Meier), usw.

Hierarchische Datenbanken sind relativ unflexibel und lassen Strukturänderungen nur mit erheblichem Aufwand realisieren. Der hierarchische Aufbau ist auch grundsätzlich ungeeignet zum Modellieren von Systemzusammenhängen. Informationen weisen häufig Bezüge zu verschiedenen hierarchischen Ebenen auf und würden daher in einer rein hierarchischen Gliederung eine redundante Speicherung erfordern. Die hierarchischen Datenbanken sind daher heute weniger häufig zu finden als relationale Datenbanken.

Relationale Datenbanken verwalten die Daten thematisch geordnet in Tabellen. Die Tabellen können miteinander in Beziehung gesetzt werden. Strukturerweiterungen können durch Integration zusätzlicher Tabellen leichter realisiert werden als bei hierarchischen Datenbanken. Ein Vorteil relationaler Datenbanken ist somit ihre größere Flexibilität. Relationale Datenbanken haben andererseits die Nachteile, daß die Abfragegeschwindigkeit relativ langsam ist und daß sie bei komplexen Beziehungen mit zahlreichen Tabellen schnell unübersichtlich werden. Weitere Nachteile relationaler Datenbanken sind die mangelhafte Integration von Funktionen und Schwächen bei der Verarbeitung multimedialer Daten, weil relationale Datenbanken vorrangig auf die Verarbeitung schriftlicher, formaterichter Daten ausgelegt sind. Problematisch ist auch die zweidimensionale Speiche-

rungsstruktur der Daten in Zeilen und Spalten.

Relationale Datenbanken dienen überwiegend der Aufnahme strukturierter, sogenannter harter Daten. Darunter versteht man Daten mit einem festen Datentyp (z. B. Zahl, Text, Währung etc.) und einer definierbaren Feldlänge. Für unstrukturierte, weiche Daten werden z. B. bei Microsoft Access® sogenannte Memofelder bereitgestellt, die größere Datenmengen aufnehmen können.

Weitere Informationen über die Grundlagen relationaler Datenbanken findet man z. B. in Meier, Andreas: Relationale Datenbanken: Eine Einführung für die Praxis; 3. Auflage, Berlin Heidelberg (Springer 1998).

Im Bereich komplexerer Anwendungen mit multimedialen Funktionen stoßen relationale Datenbanksysteme an ihre Leistungsgrenzen. Für solche Anwendungen wurden daher objektrelationale und objektorientierte Datenbanksysteme entwickelt. Objektrelationale Datenbanksysteme (ORDBS) verwenden ebenso wie die relationalen Datenbanksysteme Tabellen als Strukturierungsmittel. Struktur und Datenbanksprache wurden um objektorientierte Technologien erweitert, so daß sich in den Tabellen beliebige komplexe Objekte befinden können, die durch Datenbankoperationen ausgelöst werden können. Ein Objekt ist allgemein ein Gegenstand des Interesses, insbesondere einer Beobachtung, Untersuchung oder Messung. Objekte könnte demnach z. B. eine Person, eine Aussage, eine Sache oder ein Meßwert sein.

Klassen fassen Objekte mit gleichen Eigenschaften zusammen (z. B. kann die Klasse Personen mit der Unterkategorie juristische Personen definiert werden).

Reine objektorientierte Datenbanksysteme (OODBS) bauen im Gegensatz zu den objektrelationalen Datenbanksystemen nicht mehr auf Tabellen auf. Die Objekte werden direkt auf der Ebene der Programmiersprache definiert, was Vorteile bei Hypermediaanwendungen, CAD-Anwendungen sowie wissenschaftlich-technischen Anwendungen hat. Ein Nachteil der reinen objektorientierten Datenbanksysteme ist die mangelnde Datenunabhängigkeit.

Die Erfindung basiert auf dem System der relationalen Datenbanken. Im folgenden sollen daher einige Grundbegriffe relationaler Datenbanken weiter erläutert werden.

Fig. 1 zeigt ein Beispiel einer Tabelle einer relationalen Datenbank. In der Datenbanktheorie und der Informatik werden heute unterschiedliche Bezeichnungen für gleiche Gegenstände verwendet. Die Tabelle als grundlegendes Strukturierungsmittel wird auch als Relation bezeichnet, und die Spaltenbezeichnungen der Tabelle werden Attribute genannt. Das Attribut Name hat für die Person mit der Personenidentifikation 2 in der Tabelle z. B. den Attributwert Huber. Die Anzahl der Spalten wird als Grad bezeichnet. Ein Element ist ein bestimmtes, eindeutig identifizierbares Objekt der realen Welt oder der Vorstellung (z. B. ein Wort, eine Aussage, eine Person, eine konkrete Sache etc.). Elemente des gleichen Typs können in Elementmengen oder Elementtypen klassifiziert werden. Die Elementmengen entsprechen den Klassen in der objektorientierten Softwareentwicklung.

Jeder Datensatz einer Tabelle wird über einen Identifikationschlüssel – in der Regel eine frei definierbare Zahl oder ein Autowert als Attribut – innerhalb der Tabelle eindeutig identifiziert. Eine doppelte Vergabe eines Identifikationschlüsselwertes innerhalb einer Tabelle ist nicht zulässig und wird zu einem Fehler führen. Der Identifikationschlüssel kann auch durch eine eindeutige Kombination mehrerer Attribute hergestellt werden. In diesem Falle sollte die Kombination minimal sein. Der Identifikationschlüssel repräsentiert einen Tupel (Datensatz eines Elementes) und dient zur Herstellung von Beziehungen zwischen Tabellen.

Die Beziehung zwischen Tabellen werden dadurch herge-

3  
stellt, daß ein Identifikationsschlüssel einer Tabelle 1 als so genannter Fremdschlüssel in einer Tabelle 2 verwendet wird. Der Fremdschlüssel wird somit definiert als ein Attribut, das in einer Tabelle 2 enthalten ist und in einer Tabelle 1 den Identifikationsschlüssel bildet.

Die Beziehungsdefinitionen legen fest, wie oft ein Tupel der Tabelle 1 einem Tupel der Tabelle 2 zugeordnet werden kann. Es sind 1 : 1-, 1 : n-, n : 1- und n : m-Beziehungen möglich.

Die 1 : 1-Beziehung bedeutet, daß genau ein Datensatz aus der Tabelle 1 einem anderen Datensatz aus der Tabelle 2 zugeordnet wird. Die 1 : n-Beziehung bedeutet, daß aus einer Tabelle 1 beliebig oft auf ein Element einer Tabelle 2 Bezug genommen werden kann. Zum Beispiel können einem Element aus einer Tabelle "Person" beliebig viele Elemente aus einer Tabelle "Bankverbindung" zugeordnet werden. Eine n : 1-Beziehung ist die Umkehrung der 1 : n-Beziehung.

Ferner gibt es bei relationalen Datenbanken n : m-Beziehungen, wie sie z. B. bei der Verbindung von Personen und Projekten vorstellbar sind, wenn eine Person als Auftraggeber vieler Projekte auftritt, während in einem einzelnen Projekt nicht nur eine, sondern mehrere Personen beteiligt sind. Solche n : m-Beziehungen zwischen zwei Tabellen (im vorliegenden Beispiel Personen und Projekte) sind nicht als einfache Verknüpfungen darstellbar, weil dies zu keinem eindeutigen Ergebnis führen würde. Eine n : m-Beziehung wird daher hergestellt, indem eine weitere Tabelle eingefügt wird, eine sogenannte Beziehungstabelle, welche die n : m-Beziehung in zwei 1 : n-Beziehungen aufspaltet. Jeder Datensatz in dieser Beziehungstabelle wird durch einen zusammen gesetzten Identifikationsschlüssel eindeutig definiert, der aus Fremdschlüsseln besteht, welche auf den Identifikationsschlüsseln in den zu verknüpfenden Tabellen beruhen.

Ausgehend von diesem Stand der Technik ist es eine Aufgabe der Erfindung, ein System und ein Verfahren zum Aufbauen einer Datenbank anzugeben, die ein flexibles und anpassungsfähiges Datenbankmodell ermöglichen, welches eine minimale Redundanz bei der Datenspeicherung erfordert.

Diese Aufgabe wird durch ein rechnergestütztes System zum Aufbauen einer Datenbank mit den Merkmalen von Anspruch 1 sowie durch Verfahren zum Aufbauen eines Datenbanksystems mit den Verfahrensschritten von Anspruch 13 bzw. 15 gelöst. Die Erfindung setzt eine neue Art der Definition, Organisation, Strukturierung und Verknüpfung von Speicherbereichen in Form von Tabellen ein, um ein besonders flexibles, erweiterbares Datenbankmodell aufzubauen. Die Steuerung der Datenspeicherung, -wiedergewinnung, -zuordnung etc. kann über einen üblichen Prozessor erfolgen.

Die Erfindung beruht auf den Modellen für relationale Datenbanksysteme, sie löst sich jedoch von dem Konzept der Verwendung von inhaltlichen Attributen in den Grundtabellen (z. B. Nachname, Vorname etc. als Attribute der Tabelle "Natürliche Personen") und stützt sich statt dessen auf die Speicherung und Wiedergewinnung von Informationen in Beziehungen zwischen den Tabellen, wobei die Beziehungsdatensätze ihrerseits in Tabellen gespeichert sind. Dabei beruht die Erfindung auf neurobiologischen Erkenntnissen zur Funktionsweise des menschlichen Gehirns, Wissen in Beziehungen zu speichern, sowie auf Grundsätzen der Semiotik, gemäß denen die Bezeichnung und das Bezeichnete getrennt werden. In Anlehnung an die Neurobiologie könnte man bei der Erfindung von einem synaptischen Modell in Abgrenzung zum bisher üblichen attributiven Modell sprechen.

Die Erfindung sieht Grundtabellen für zu jeweils einer Grundklasse gehörende Elemente vor. Bei einer bevorzug-

ten Ausführungsform der Erfindung orientieren sich diese Grundtabellen inhaltlich an den Grundmodellen menschlichen Wissens und werden nach Erkenntnissen der Neurobiologie zunächst in die Klassen: Bewegender, Bewegliches

5 und Lokalisierung – oder Personen, Sachen und Orte eingeteilt. Weitere Grundklassen ergeben sich aus der Semiotik als Wörter, Bedeutungen und Bilder, um die in den ersten drei Klassen beschriebene reale Welt von der Wahrnehmung und Bezeichnung dieser realen Welt trennen zu können und

10 so einen möglichst hohen Abstraktionsgrad des Datenbankmodells zu erreichen. Schließlich werden zur Darstellung der Organisation der komplexen Welt noch zusätzliche künstliche Ordnungsobjekte als Grundklassen vorgesehen, nämlich Projekte, Vorgänge (zur Darstellung der Zeit) und

15 Objekte.

Hieraus ergibt sich das in Fig. 2 dargestellte Wissensmodell, in dem die Grundklasse "Hyperlinks" stellvertretend für die Sinneswahrnehmungen Sehen und Hören oder in der praktischen Umsetzung die Verwaltung von Bildern und anderen Dateien steht.

Mit dem erfindungsgemäßen System können die in Fig. 3 dargestellten Beziehungen zwischen den Grundklassen hergestellt werden.

Die dargestellten Grundklassen gehören zu dem bevorzugten Ausführungsbeispiel der Erfindung, wobei die Erfindung auf diese Grundklassen weder festgelegt noch beschränkt ist, es können also einige oder alle der beispielhaft genannten Grundklassen fehlen bzw. durch andere Klassen ersetzt und/oder erweitert werden. Die Verwendung von Hyperlinks verweist, wie gesagt, auf externe Dateien sowie in Dateiform verwaltete Inhalte audiovisueller Art und stellt einen Kompromiß im Rahmen der heute praktikablen Technologie dar.

Die Elemente der jeweiligen Grundtabellen werden über 35 wenigstens eine vorzugsweise mehrere Beziehungstabellen miteinander verknüpft. Dabei kann jedes Element einem anderen zugeordnet werden, einschließlich einem anderen Element innerhalb derselben Tabelle. Ein Überblick über die möglichen Beziehungen in dem erfindungsgemäßen Datenbanksystem gibt Fig. 4.

Die doppelte Zuordnung kann anhand der folgenden Beispiele besser verstanden werden: ein Wort kann einer Person zugeordnet werden, die dadurch entstehende Beziehung Wort-Person kann z. B. als "Nachname" definiert werden. 45 Die Person kann aber auch einem Wort zugeordnet werden, wobei die Beziehung Person-Wort z. B. zwischen einer Person P und dem Wort Dampfmaschine bestehen und als "Erfinder" definiert sein könnte. Diese bidirektionale Zuordnung gilt auch für die Verknüpfung gleicher Elemente, wo 50 bei sich dann die Darstellung der Fig. 4a ergibt.

Während die Beziehungen zwischen den Elementen der Grundklassen an sich durch die Beziehungstabellen hergestellt werden, sieht die Erfindung zusätzlich noch eine Kategorietabelle vor, die mit der Beziehungstabelle verbunden wird, um den jeweiligen Beziehungen eine eindeutige Beziehungskategorie, wie Nachname oder Erfinder in dem obigen Beispiel, zuzuordnen.

Durch den erfindungsgemäßen Aufbau des Datenbanksystems mit mehreren Grundtabellen, vorzugsweise mehreren Beziehungstabellen und einer Kategorietabelle erhält man ein äußerst flexibles Datenbankmodell, das ein Minimum an Redundanz erzeugt. Während im Stand der Technik die Kategorie einer Beziehung und die Beziehung selbst durch attributive Zuweisung innerhalb einer Tabelle untrennbar verknüpft waren, löst die Erfindung diese Verknüpfung auf und erhöht damit die Flexibilität des Datenbanksystems bei gleichzeitiger Verringerung der Redundanz. Die einzelnen Grundtabellen und die ihnen zugewiesenen Speicherberei-

che enthalten somit in der Regel weniger Information als bei Datenbanken des Standes der Technik, weil ein wesentlicher Teil der Information in die Beziehungstabellen ausgelagert ist, welche ihrerseits auf weitere Informations- und die Kategorietabelle zugreifen. Einem vollständigen Datensatz für eine Abfrage erhält man somit erst durch die Verknüpfung mehrerer Tabellen. Dadurch können die Strukturen der einzelnen Tabellen einfacher und übersichtlicher gehalten werden. Ein wesentlicher Aspekt der Erfindung ist, daß unabhängig von der inhaltlichen Klassenbezeichnung der Grundtabellen zur Definition eines Beziehungsdatensatzes zwischen zwei Grundtabellen oder zwei Elementen aus einer Grundtabelle immer nur auf eine einzige Kategorietabelle zurückgegriffen wird.

Das erfindungsgemäße Datenbanksystem ist in eine inhaltliche Ebene und in eine Meta-Ebene gegliedert. Zur inhaltlichen Ebene gehören die Grundtabellen und die Beziehungstabellen, wobei für die Verknüpfung von jeweils zwei Grundtabellen eine Beziehungstabelle vorgesehen werden kann; zur Meta-Ebene gehört zunächst die Kategorietabelle, welche den jeweils in der Beziehungstabelle enthaltenen Beziehungen eine Kategorie (Beziehungsdatensätzen) zuordnet, z. B. einer Beziehung (= Beziehungsdatensatz) Wort-Person die Kategorie "Nachname" oder einer Beziehung (einem Beziehungsdatensatz) Person-Wort die Kategorie "Erfinder".

Bei einer bevorzugten Ausführungsform der Erfindung enthält die Meta-Ebene ferner eine Relationstabelle, die alle möglichen Beziehungen zwischen den Tabellen enthält und mit der Kategorietabelle verbunden wird, um jeweils einer Beziehungskategorie, wie der Kategorie "Name", mögliche Beziehungen zwischen Grundtabellen zuzuordnen. So ist es z. B. sinnvoll der Beziehungskategorie "Name" die möglichen Beziehungen Person-Wort, Sache-Wort, Projekt-Wort und Objekt-Wort zuzuordnen (während beispielsweise eine Zuordnung der Beziehung Ort-Zeit zu der Kategorie "Name" unsinnig wäre). Die Relationstabelle dient somit der besseren Strukturierung und Übersichtlichkeit der Datenbank.

Bei der bevorzugten Ausführungsform wird ebenfalls in der Meta-Ebene eine Kategorie-Relationstabelle vorgesehen, welche die Kategorietabelle mit der beschriebenen Relationstabelle verbindet und eine Beziehung einschließlich einer Beziehungsrichtung für die Verknüpfung jeweils zweier Elemente kennzeichnet.

Gemäß eines weiteren Aspekts der Erfindung wird in der Meta-Ebene eine Untertabelle für eine Untergruppe von zu einer Grundklasse gehörenden Elementen vorgesehen, die über eine Beziehungs-Untertabelle mit der Kategorietabelle verbindbar ist, um Beziehungskategorien zu bestimmen, die der Untergruppe zugeordnet werden sollen. Diese Untertabellen dienen dem Zweck, für eine Grundklasse "Person" beispielsweise die Unterklassie "natürliche Person" zu definieren. Der Unterklassie "natürliche Person" können dann bestimmte Kategorien zugeordnet werden, wie Nachname, Vorname, Geburtsdatum etc., um aufgrund dieser zu einer Unterklassie gehörenden Kategorien für die Ein- und Ausgabe in das Datenbanksystem ein dynamisches Formular zu erzeugen, welches nur die Beziehungskategorien der Unterklassie enthält. Weiterhin sieht das erfindungsgemäße System eine Aktionstabelle für Eingabe/Ausgabe-Aktionen zur Erstellung und Nutzung der Datenbank vor.

Die Erfindung ist im folgenden anhand einer bevorzugten Ausführungsform mit Bezug auf die Zeichnungen näher erläutert. In den Figuren zeigen:

**Fig. 1** eine Grafik zur Erläuterung einiger in der Datenbanktheorie verwendeten Begriffe;

**Fig. 2** eine Darstellung eines Wissensmodells zur Ablei-

tung wichtiger Grundklassen für das erfindungsgemäße Datenbanksystem;

**Fig. 3** eine graphische Darstellung der möglichen Beziehungen zwischen den Grundklassen, die in **Fig. 2** gezeigt sind;

**Fig. 4** eine andere Art der Darstellung der möglichen Beziehung zwischen den Grundklassen der **Fig. 2**;

**Fig. 4A** eine ähnliche Darstellung wie in **Fig. 4** unter Einschluß aller bidirektionalen Beziehungen;

**Fig. 5A–5I** schematische Darstellungen der Strukturen verschiedener Grundtabellen für das erfindungsgemäße Datenbanksystem;

**Fig. 6A–6F** Beispiele für die Strukturen von Beziehungstabellen gemäß der Erfindung;

**Fig. 7** ein Beispiel für eine Struktur der Kategorietabelle gemäß der Erfindung;

**Fig. 8A–8J** Beispiele für die Strukturen verschiedener Tabellen der Meta-Ebene;

**Fig. 9A–9C** schematische Darstellungen der Verknüpfungen ausgewählter Grundtabellen mit anderen Grundtabellen des erfindungsgemäßen Datenbanksystems; und

**Fig. 10** eine schematische Darstellung der Meta-Ebene des erfindungsgemäßen Datenbanksystems.

Während die vorliegende Erfindung anhand bestimmter bevorzugter Grundklassen für die Grundtabellen sowie ausgewählter Beispiele für die Elemente erläutert wird, wird der Fachmann verstehen, daß sich die erfindungsgemäße Datenbankstruktur sowie die besondere Art der Verknüpfung der Elemente und Tabellen der Datenbank auch auf jede andere Datenbank mit teilweise oder vollständig anderen Grundklassen und Inhalten übertragen läßt. Um eine minimale Redundanz zu erreichen enthält das erfindungsgemäße Datenbanksystem jedoch vorzugsweise immer die Grundklasse Wort.

Folgende Abkürzungen werden in der Beschreibung und den Figuren verwendet:

tbl – Tabelle

Key – Wort

Mmo – Aussage/Memo

Hyp – Hyperlink

Prs – Person

Itm – Sache

Loc – Ort

Prc – Vorgang/Zeit

Prj – Projekt

Obj – Objekt

Cat – Kategorie

Rel – Relation

Ele – Grundklasse

SubEle – Untergruppe einer Grundklasse

Z – Meta-Ebene

ID – Identifikation

Für das System und Verfahren zum Aufbauen einer Datenbank gemäß der Erfindung gilt grundsätzlich, daß jede Grundklasse als eigenständige Tabelle definiert wird, die für jedes Element einen eindeutigen Identifikationsschlüssel oder Primärschlüssel enthält. Für die Beziehungen gilt allgemein, daß sie in Beziehungstabellen definiert werden, wobei die Identifikationsschlüssel der zugehörigen Elemente aus den Grundtabellen, zwischen denen Beziehungen hergestellt werden sollen, als Fremdschlüssel in der zugehörigen Beziehungstabelle verwendet werden. Weitere Merkmale der Beziehungen erscheinen als zusätzliche Attribute in der Beziehungstabelle.

Im folgenden werden die Grundtabellen und die Beziehungstabellen für die inhaltliche Ebene und anschließend die Tabellen der Meta-Ebene erläutert.

**Fig. 5A** zeigt eine mögliche Struktur für die Grundtabelle

tblKey der Grundklasse "Wort".

Für die zentrale Verwaltung von Worteinheiten in "atomarer" Form wurde die englische Bezeichnung Keywords gewählt, da die Wörter quasi den Schlüssel zur Information darstellen.

Dabei ist auf eine Trennung der Objekte, ihrer Bezeichnungen und der Bedeutung der Zeichen zu achten, wobei den Wörtern die Funktion der Bezeichnung zukommt. Entsprechend diesem Ansatz werden die als Worte vorkommenden Zeichen in einer zentralen Tabelle tblKey erfaßt und zu den bezeichneten Objekten jeweils über eine Kategorie (z. B. Nachname in der Person-Wort-Beziehung) in Beziehung gebracht.

Als Identifikationsschlüssel dient das Attribut KeyID (Autowert; Datentyp Long Integer). Über das Attribut KeyName wird das eigentliche Wort erfaßt (Datentyp Text, bis zu 255 Zeichen). Das Attribut KeyRank dient dazu, dem Wort (bzw. Keyword oder Schlüsselwort) einen numerischen Wert zuzuordnen, um eine von der alphabetischen Reihenfolge abweichende Sortierung zu ermöglichen (Datentyp Zahl).

Fig. 5B zeigt eine mögliche Struktur für die Grundtabelle tblMmo der Grundklasse "Aussage".

Die Bedeutung von Zeichen wird zwar wiederum in Zeichen zum Ausdruck gebracht, jedoch in ausführlicherer Form von Aussagen, d. h. in unter Umständen längeren Textpassagen erklärenden Charakters. Für die Administration unstrukturierter Daten (sog. soft data) wie Ideen, Aufgaben, Sachverhalte (casebooks, casestudies) wurde daher der Komplex Aussage/Memo entwickelt.

Da das Datenbankverwaltungssystem von Microsoft Access®, das dem Erfinder für die Entwicklung einer Ausführungsform seiner Erfahrung diente, ein eigenes Format zur Verwaltung großer Textmengen in sog. Memo-Feldern vorsieht, erhält die Tabelle zur Aufnahme von Aussage/Memo-Daten den Namen tblMmo.

Identifikationsschlüssel der Tabelle tblMmo ist das Attribut MmoID (Autowert; Datentyp Long Integer). Das Attribut zur Aufnahme der Aussage/Memo-Daten wurde Mmo genannt (Datentyp Memo). Die Attribute MmoCreated und MmoChanged sind Datumsfelder zur Speicherung des Datums der Erzeugung des Memos und der letzten Änderung.

Fig. 5C zeigt eine mögliche Struktur für die Grundtabelle tblHyp der Grundklasse "Hyperlink".

Die Anbindung an externe Dateien erfolgt über Hyperlinks. Die Tabelle zur Verwaltung dieser zentralen Kommunikationsobjekte wurde daher tblHyp genannt.

Die Tabelle tblHyp besitzt den Identifikationsschlüssel HypID (Autowert; Felddatentyp Long Integer). Das Attribut Hyperlink dient zur Aufnahme des Hyperlinks und besitzt den Felddatentyp Hyperlink, so daß Hyperlink-Funktionen direkt aus der Datenbank ausgeführt werden können. Das Attribut HypDCC (Felddatentyp Text) dient zur Aufnahme eines Hyperlink-Duplizitäts-Kontroll-Codes.

Fig. 5D zeigt eine mögliche Struktur für die Grundtabelle tblPrs der Grundklasse "Person".

Für die Darstellung von Personen und Organisationen wurde in Anlehnung an die Bezeichnung Persons bzw. Personen die Tabelle tblPrs vorgesehen.

Die Tabelle tblPrs besitzt die Attribute PrsID (Identifikationsschlüssel, Autowert; Felddatentyp Long Integer) sowie die Attribute PrsCreated und PrsChanged (Felddatentyp Datum) zur Speicherung des Anlagedatums des Datensatzes und des Datums der letzten Änderung. Das Attribut PrsDCC dient zur Aufnahme eines Personen-Duplizitäts-Kontroll-Codes. Alle weiteren – herkömmlich als Attribute konzipierten Informationen zu einer Person, (z. B. Nachname, Vorname etc.) werden funktional über die Beziehung zu je

einer anderen Grundtabelle des inhaltlichen Modells und die Beziehungsdefinition mit Hilfe der Kategorietabelle hergestellt. (wie noch erläutert wird).

Fig. 5E zeigt eine mögliche Struktur für die Grundtabelle 5tblItm der Grundklasse "Sache".

Einzelne, konkrete Gegenstände (z. B. das konkrete Buch in der Bibliothek mit einer Archivierungsnummer im Gegensatz zum zugrundeliegenden abstrakten Publikationsprodukt mit einer ISBN-Nummer) werden in Anlehnung an die 10 englische Bezeichnung Items in einer Tabelle tblItm repräsentiert.

Fig. 5F zeigt eine mögliche Struktur für die Grundtabelle 15tblLoc der Grundklasse "Ort".

Die Orte, an denen Personen wohnen, Projekte stattfinden, einzelne Gegenstände sich befinden werden in der Tabelle 15tblLoc (für Locations) repräsentiert.

Fig. 5G zeigt eine mögliche Struktur für die Grundtabelle 20tblPrj der Grundklasse "Projekt".

Für Projekte (Projects) als zielgerichtete Vorhaben bestimmter Personen oder Organisationen wurde die Tabellenbezeichnung 20tblPrj gewählt.

Fig. 5H zeigt eine mögliche Struktur für die Grundtabelle 25tblObj der Grundklasse "Objekt".

Die übrigen abstrakten Objekte – Produkte, Artikelbezeichnungen, Gegenstände im weiteren abstrakten Sinne, Marken, Arten, Gattungen – werden als Objects in der Tabelle 25tblObj repräsentiert.

Die Tabellen 30tblItm, 30tblLoc, 30tblPrj und 30tblObj sind nach demselben Prinzip wie die Tabelle 30tblPrs aufgebaut. Identifikationsschlüssel sind die Attribute Itma für die Tabelle

30tblItm, Loc-ID für die Tabelle 30tblLoc, PrjID für die Tabelle 30tblPrj und ObjID für die Tabelle 30tblObj (jeweils Autowert, Felddatentyp Long Integer). Die einzigen weiteren Attribute der Tabellen sind Datumsfelder zur Speicherung der Anlage und der letzten Änderung der Datensätze. Das Textattribut ...DCC dient wiederum zur Aufnahme eines Duplizitäts-Kontroll-Codes für die bezeichneten Elemente.

Fig. 5I zeigt eine mögliche Struktur für die Grundtabelle 35tblPrc der Grundklasse "Zeit".

Die Einbindung von Personen, Projekten, Produkten und Objekten in die Zeit erfolgt über Vorgänge, deren englische Bezeichnung als Procedures bzw. Proceedings erfolgen kann. Die "Vorgangstabelle" wurde daher mit der Bezeichnung 40tblPrc versehen.

Identifikationsschlüssel der Tabelle 45tblPrc ist das Attribut PrcID. Weitere Attribute sind die Datumsfelder PrcStart und PrcEnd zur Speicherung des Beginns und des Endes eines Vorgangs. Das Attribut PrcDCC dient zur Aufnahme eines Duplizitäts-Kontroll-Codes für Vorgänge. Im übrigen sind 50 auch hier die weiteren Informationen nicht in dem Tupel der Grundtabelle enthalten, sondern in über die Kategorietabelle definierten Beziehungen zu anderen Tupeln gespeichert.

Datenbanktechnisch bestehen zwischen diesen Systemgrundbestandteilen, nämlich den Grundtabellen, sog. n : m-Beziehungen, d. h., daß z. B. beliebig viele Personen mit beliebig vielen Projekten, Produkten, Objekten, Orten und Vorgängen in Beziehung stehen können.

Unter den Elementen der Grundtabellen bestehen 45 Beziehungsmöglichkeiten, wenn man davon ausgeht, daß eine 60 Grundklasse mit sich selbst verbunden sein kann. Dies ist aufgrund des hohen Abstraktionsgrades der Grundklassen häufig der Fall. Z. B. kann eine natürliche Person P1 mit der Gesellschaft G in Beziehung stehen (z. B. als Geschäftsführer). Beide werden unter der Objektklasse Personen verwaltet. Daher müssen Beziehungstypen vorgesehen werden, über die die jeweilige Grundklasse mit sich selbst verbunden werden kann (hier also als Person-Person-Beziehung), wie in Fig. 4 gezeigt.

Die Benennung der Beziehungstabellen wurde hier so vorgenommen, daß zum Präfix *tbl* jeweils eine Kürzelkombination der in Beziehung stehenden Grundklassen gewählt wird (die Reihenfolge ist – wie noch zu zeigen sein wird – unerheblich). Für die Beziehung von Personen zu Projekten wird also z. B. der Präfix *tbl* mit der Kombination aus den Kürzeln *Prs* und *Prj* versehen, so daß die Beziehungstabelle *tblPrsPrj* genannt wird (oder bedeutungsgleich *tblPrj Prs*).

**Fig. 6A** zeigt eine mögliche Struktur der Beziehungstabelle *tblKeyKey* für die Beziehung "Wort-Wort".

Die Tabelle *tblKeyKey* dient zur Darstellung der rekursiven Beziehungen innerhalb der Tabelle *tblKey*. Es wird ein zusammengesetzter Identifikationsschlüssel aus den Fremdschlüsseln der Tabellen *tblKey* und *tblCat* (Kategorietabelle, die noch erläutert wird) gebildet. Aus der Tabelle *tblKey* wird dabei der Identifikationsschlüssel sowohl für das Schlüsselattribut *Keya* als auch für das Schlüsselattribut *Key1ID* als Fremdschlüssel entnommen. Da zwei identische Wortkombinationen in unterschiedlicher Funktion aufeinander bezogen sein können, ist die Eindeutigkeit der Datensätze erst hergestellt, wenn auch die Kategorie als Fremdschlüssel aus *tblCat* in den zusammengesetzten Identifikationsschlüssel einbezogen wird.

Als Beispiel können die Worte "Fundament" und "fundamentum" dienen; "fundamentum" ist sowohl Herkunftswoort als auch deutsch-lateinische Übersetzung zum Wort "Fundament". Will man beide Kategorien im Datenbanksystem abbilden, muß wegen der Anforderung der Eindeutigkeit der Datensätze die jeweilige Kategorie (also Herkunftswoort und deutsch-lateinische Übersetzung) in den zusammengesetzten Identifikationsschlüssel integriert werden.

Der Name des Identifikationsschlüssels *CatID* wird dabei in den Beziehungstabellen, in denen der Identifikationschlüssel aus *tblCat* als Fremdschlüssel verwendet wird, um den Bezeichnungsnamen ergänzt. Der Fremdschlüssel wird somit im Beispiel der Wort-Wort-Beziehung nicht *CatID*, sondern *CatKeyKeyID* genannt.

Das Attribut *KeyKey123* dient der Sortierung und ist vom Felddatentyp Zahl, das Attribut *KeyKeyTxt* ist für kurze Anmerkungen und Notizen gedacht, das Attribut *KeyKeyActivate* ist ein Ja/Nein-Wert und dient der gezielten Unterdrückung einzelner Datensätze der Beziehungsmenge bei der Datenausgabe.

Nach dem Grundmuster der Tabelle *tblKeyKey* sind so gut wie alle Beziehungstabellen aufgebaut.

**Fig. 6B** zeigt eine mögliche Struktur der Beziehungstabelle *tblKeyMmo* für die Beziehung "Wort-Aussage".

**Fig. 6C** zeigt eine mögliche Struktur der Beziehungstabelle *tblKeyHyp* für die Beziehung "Wort-Hyperlink".

Die übrigen Beziehungsmengen, an denen Keywords (Wörter) beteiligt sind, sind sprachlich so formuliert, daß das Kürzel "Key" nachgestellt ist. Diese Ordnung ist rein sprachlicher Natur, da die Beziehungsrichtung im Rahmen der Kategoriedefinition festgelegt wird.

**Fig. 6D** zeigt eine mögliche Struktur der Beziehungstabelle *tblMmoMmo* für die Beziehung "Aussage-Aussage".

In der Tabelle *tbMmoMmo* werden die rekursiven Beziehungen von Memodatensätzen verwaltet. Dies betrifft zum Beispiel den Fall, daß eine Notiz einer anderen Notiz als Folgedanke zugeordnet wird. Auch hier setzt sich der Identifikationsschlüssel zusammen aus den Fremdschlüsseln der Tabelle *tblMmo* sowie der Tabelle *tblCat*. Hinsichtlich der Attribute *MmoMmo123*, *MmoMmoTxt* und *MmoMmoActivate* kann auf die Angaben zu *TblKeyKey* verwiesen werden.

**Fig. 6E** zeigt eine mögliche Struktur der Beziehungstabelle *tblMmoHyp* für die Beziehung "Aussage-Hyperlink".

Die übrigen Beziehungsmengen, an denen Memos (Aus-

sagen) beteiligt sind, sind entsprechend aufgebaut, wobei wie oben darauf hinzuweisen, daß die Ordnung rein sprachlicher Natur ist, da die Beziehungsrichtung ausschließlich im Rahmen der Kategoriedefinition festgelegt wird (s. unten Meta-Ebene).

**Fig. 6F** zeigt eine mögliche Struktur der Beziehungstabelle *tblHypHyp* für die Beziehung "Hyperlink-Hyperlink".

Die gezeigten Strukturen der Beziehungstabellen dienen lediglich der Erläuterung der Erfindung, wobei die hier nicht explizit dargestellten Beziehungstabellen in der Regel entsprechend aufgebaut sein werden, die Erfindung jedoch nicht auf das für dieses Ausführungsbeispiel gewählte Format beschränkt ist.

Hinsichtlich der Attribute zur Verwaltung der Kategorien ist darauf hinzuweisen, daß die Fremdschlüsselattribute (z. B. *CatHypHypID*, *CatMmoHypID* etc.) alle auf den Identifikationsschlüssel *CatID* aus der Tabelle *tblCat* Bezug nehmen, jedoch aus Gründen der Übersichtlichkeit in den Beziehungsmengen nicht mit der einheitlichen Attributbezeichnung *CatID*, sondern zusätzlich mit dem jeweiligen Beziehungstyp versehen wurden.

Neben der inhaltlichen Ebene des erfindungsgemäßen Datenbanksystems, welche die Grundtabellen und die Beziehungstabellen enthält, steht die Metalebene, die für die übergeordnete Organisation der Datenbank zuständig ist.

Die zentrale Stellung in der Metalebene kommt den Kategorien zu, über welche die Beziehungen der verschiedenen Elemente zueinander definiert werden. Dies soll am Beispiel der Grundklassen Personen, Wörter und Sachen näher erläutert werden. Personen werden üblicherweise Attribute, wie Namen, Vornamen, Rechtsform etc., zugeordnet. In der Realität existiert ein Name, z. B. "Zimmermann", jedoch unabhängig von der betreffenden Person als Begriff des allgemeinen Wortschatzes. In seiner Beziehung zu Personen kann diesem Begriff eine Doppelfunktion zukommen: er kann sowohl Namen als auch Berufsbezeichnung einer Person darstellen. Die üblicherweise als Attribute konstruierten Kategorien "Name" und "Beruf" können bei der Erfindung auch funktional als ein Bindeglied zwischen einer Person und einem Wort des allgemeinen Wortschatzes verwendet werden. Die gleich Konstellation läßt sich noch deutlicher am Beispiel der Beziehungen der Grundklassen Personen und Sachen darstellen. Man könnte einer Sache das Attribut "Eigentümer" zuweisen und dann eine Person als Eigentümer eintragen. Damit würde die Person der Sache wie eine Eigenschaft anhaften, und ebenso das Eigentum der Person. Tatsächlich definiert das Eigentum jedoch eine Funktion zwischen einer Person und einer Sache. Gleichzeitig ist das Eigentum als Begriff Bestandteil des allgemeinen Wortschatzes und als solcher auch in anderem Kontext verwendbar. Dieses Verhältnis von Objekten, Wörtern, Begriffen und Funktionsbezeichnungen (Kategorien) ist bei der Erfindung auf alle Systembestandteile übertragen.

**Fig. 7** zeigt eine mögliche Struktur der Kategorietabelle *tblZCat*.

In der Tabelle *tblZCat* (für Categories) werden die Informationskategorien verwaltet, so daß z. B. die Informationskategorie Autor nur ein einziges Mal im System verwaltet werden muß. Die Informationskategorien werden im Attribut *ZCat* (Feldtyp Text) verwaltet. Das Attribut *ZCatRnk* (= Kategorien-Ranking) dient dazu, einer Kategorie einen numerischen Wert für Sortierungszwecke zuzuweisen. Soll z. B. in einem Datenbankreport die Kategorie Titel vor der Kategorie Autor angezeigt werden, kann der Kategorie Titel z. B. der numerische Wert 1 und der Kategorie Autor der numerische Wert 2 zugewiesen werden. Der Feldtyp des Attributs *ZCatRnk* ist Zahl (Long Integer).

**Fig. 8A** zeigt eine mögliche Struktur der Relationstabelle

## tblZRel.

Die oben verschiedenen Beziehungs(richtungs)möglichkeiten werden in einer eigenen Tabelle tblZRel verwaltet. Das Tabellenkürzel Rel steht für Relationships. Das zwischen Präfix und Tabellenkürzel gestellte Z dient der alphabetischen Sortierung der Metatabellen am Ende aller verwalteten Tabellen. Jeder Metatabelle wird daher ein Z vor das Tabellenkürzel zugeordnet.

Durch die zentrale Verwaltung aller im System vorkommenden Informationskategorien in einer Tabelle könnte man bei den üblicherweise mehreren hundert Kategorien schnell den Überblick verlieren, in welchen Beziehungen des Datenbanksystems sie relevant sind. So kommt die Kategorie "Definition" möglicherweise ausschließlich in der Beziehung Wort-Aussage/Memo vor. Es werden daher die möglichen Beziehungstypen (bei dem gezeigten Ausführungsbeispiel  $9 \times 9 + 9$  mögliche, undefinierte (d. h. nicht einer Kategorie zugeordnete) Beziehungen, z. B. PRS-ITM, ITM-PRS, ITM-ITM1; ITM1-ITM etc.) in einer eigenen Tabelle verwaltet und mit den sie betreffenden Informationskategorien verknüpft. Es kann dann eine vollständige Strukturierung der Beziehungskategorien nach den Beziehungen und Beziehungsrichtungen erfolgen, in denen die Beziehungskategorien eine Rolle spielen. Die Beziehungskategorie Definition wird dann z. B. mit der Beziehung KEY-MMO (= Wort-Aussage/Memo) verknüpft und steht dann auch nur in dieser Beziehung zur Verfügung.

Die Tabelle tblZRel (für Relationships/Associations bzw. Beziehungsrichtungen) hat als Identifikationsschlüssel das Attribut ZRelID (Autowert; Long Integer). Das Attribut ZRel (Felddatatype Text) dient zur Aufnahme der Kurzform der Beziehungsrichtung, also z. B. ITM-PRS für die Beziehung, innerhalb derer eine oder mehrere Personen einer oder mehreren Sachen zugeordnet werden können, oder PRS-ITM für die Beziehung, in der eine oder mehrere Sachen einer oder mehreren Person zugeordnet werden können.

Jeder Beziehungsrichtung kann dabei eine bestimmte sprachliche Formulierung zugeordnet werden. So kann z. B. die Beziehung ITM-PRS sprachlich (hier englisch) formuliert werden als Person is a... (Person ist zugeordnet als, z. B. als Eigentümer). Umgekehrt kann für die Beziehungs PRS-ITM die sprachliche Formulierung Item is a... (Sache ist zugeordnet als, z. B. Eigentum) definiert werden. Durch den Fremdschlüssel ZEle\_isID (Felddatatype Zahl; Long Integer) wird auf die in der Tabelle tblZEle\_is verwalteten möglichen sprachlichen Formulierungen der Zuordnung der Wissenselemente je Beziehung verwiesen.

Das Attribut ZRelT2 (Felddatatype Text) dient der sprachlichen Ausformulierung der bisher nur in Kurzform verwalteten Beziehung. Das Attribut ZRelRnk (Ranking of Relationships) dient dazu, eine von der alphabetischen Reihenfolge abweichende inhaltliche Sortierung der Beziehungstypen zu ermöglichen.

Fig. 8B zeigt eine mögliche Struktur einer Relations-Erläuterungstabelle tblZEle\_is.

Ebenfalls eine sprachlich-inhaltliche Funktion erfüllt die Relations-Erläuterungstabelle tblZEle\_is. In ihr wird die jeweilige sprachliche Formulierung einer Beziehung bzw. Beziehungsrichtung gespeichert. Dient z. B. die Beziehung Prs-Itm dazu, eine Sache einer Person zuzuordnen, dann wird die Sache als etwas zugeordnet. Der Beziehung Prs-Itm würde dann die in der Tabelle tblEle\_is verwaltete sprachliche Formulierung "Item is" zugeordnet. Das spiegelbildliche Beispiel wäre die Beziehung Itm-Prs, in der eine Person einer Sache, z. B. als Eigentümer, zugeordnet wird. Für diese Beziehung gilt die sprachliche Formulierung "Person is". Weitere Möglichkeiten sind "Hyperlink is", "Keyword is", "Location is", "Memo is", "Object is", "Person is", "Pro-

ceeding is", Project is". Für die rekursiven Beziehungen innerhalb einer Relations-Erläuterungstabelle muß eine sprachliche Formulierung hinsichtlich der Beziehungsrichtung von Attribut 2 im Verhältnis zu Attribut 1 vorgesehen werden. Dafür dienen "Item 2 is" Hyperlin 2 is", "Keyword 2 is", "Location 2 is", "Memo 2 is", "Object 2 is", "Person 2 is", "Proceeding 2 is" und "Project 2 is".

Identifikationsschlüssel der Tabelle tblZEle\_is ist das Attribut ZEle\_isID (Autowert; Long Integer). Das Attribut ZEle\_is (Felddatatype Text) dient – wie zu Tabelle tblZRel dargestellt – der Aufnahme der sprachlichen Formulierung der Assoziation (also z. B. Person is für alle Beziehungen, in denen Personen nachgestellt, also zugeordnet werden – z. B. KEY-PRS, MMO-PRS, HYP-PRS, PRS1-PRS; ITM-PRS, LOC-PRS, PRJ-PRS, PRC-PRS, OBJ-PRS).

Fig. 8C zeigt eine mögliche Struktur der Relations-Strukturierungstabelle tbIZEle.

Die Tabelle tbIZEle dient der inhaltlichen Strukturierung der in tblZRel abgebildeten Beziehungsrichtungen. In ihr werden die Grundklassen sprachlich dargestellt; also Wort, Aussage, Hyperlink, Person, Sache, Ort, Projekt, Zeit und Objekt.

Die Informationskategorien der Kategorietabelle tbIZCat sind nun zwar jeweils einer Beziehung zugeordnet, aber das Rechnersystem weiß noch nicht, welche Beziehungstypen zu welchen Grundklassen des Datenbanksystems gehören. Die Beziehungsrichtung ITM-PRS (Sachen-Personen) ist z. B. den Elementen Sachen und Personen zuzuordnen. Diese Grundklassen werden in der Tabelle tbIZEle verwaltet.

Identifikationsschlüssel ist das Attribut ZEleID (Autowert; Felddatatype Long Integer). Das Attribut ZEle (Felddatatype Text) dient zur Aufnahme des Elements (hier also der Elemente Keywords, Memos, Hyperlinks, Persons, Items, Locations, Projects, Proceedings und Objects). Mit Hilfe des Attributs ZEleRnk (Felddatatype Zahl) kann den einzelnen Elementen wiederum eine Zahl zugeordnet werden, um – wie oben – von der alphabetischen Sortierung zugunsten einer inhaltlichen Sortierung abzuweichen.

Fig. 8D zeigt eine mögliche Struktur einer Untertabelle tbIZSubEle.

Für eine bessere Übersichtlichkeit und Analysierbarkeit der Kategorien können die bisher nur in einer zentralen Kategorietabelle tbIZCat verwalteten Beziehungskategorien einer Subklasse zugeordnet werden, die hier als Subelemente der Elemente der Grundklassen bezeichnet werden. Wenn also z. B. abstrakte Objekte in Publikationen, Rechtsprechung, Normen oder dergleichen, als Subelemente untergliedert werden können, so sollte es auch möglich sein, diesen Subelementen die für sie maßgeblichen Beziehungskategorien zuzuordnen.

Die Elemente (Klassen) des erfindungsgemäßen Datenbanksystems können daher nach Bedarf in Subelemente (Subklassen) unterteilt werden. So besteht z. B. das Element (bzw. die Klasse) Personen u. a. aus den Subelementen Natürliche Personen, Gesellschaften etc. Über diese Subelemente schließt sich der Kreis zu den Beziehungskategorien: der inhaltlichen Strukturierung der Meta-Ebene dient auch die Möglichkeit, den Subelementen Beziehungskategorien zuzuordnen mit dem Ziel, z. B. sämtliche Beziehungskategorien, die einer Gesellschaft als juristische Person zugeordnet sind, abfragen zu können.

Fig. 8E zeigt eine mögliche Struktur einer Aktionstabelle tbIZAction.

In der Aktionstabelle tbIZAction werden zentral alle Aktionen bzw. Operationen verwaltet, die im Rahmen der Anwendungsprogrammierung ausgeführt werden sollen. Eine Aktion "Stiftung neu anlegen" könnte dann z. B. auf die für

Stiftungen (als Subelement zur Grundklasse Personen) maßgeblichen Beziehungskategorien abfragen und ein dynamisches Formular mit den für Stiftungen maßgeblichen Beziehungskategorien (Stiftungsname, Stiftungszweck etc.) generieren. Dafür sind geeignete Algorithmen auf Anwendungsebene vorzusehen. In der Tabelle tblZAction sollen Aktionen verwaltet werden. Über die Aktion "Stiftung neu anlegen" könnte z. B. eine Prozedur ausgelöst werden, die ein dynamisches Formular zur Verwaltung von Personen öffnet und die der Subklasse bzw. dem Subelement Stiftung zugeordneten Beziehungskategorien generiert.

Identifikationsschlüssel ist das Attribut ZActionID (Autowert; Long Integer). Die einzelnen Aktionen werden unter dem Attribut ZAction (Text) verwaltet. Das Attribut ZCatActionID (Feldtyp Zahl, Long Integer) ist Fremdschlüssel der Tabelle tblZCatAction und dient der Strukturierung in Eingabe- und Ausgabeaktionen (Input oder Output). Das Attribut ZEleID ist Fremdschlüssel der Tabelle tblZEle und dient der Zuordnung der einzelnen Aktionen zum jeweiligen Element des Datenbanksystems (z. B. Personen). Die Input-Aktion "Stiftung neu anlegen" würde also dem Element Personen zugeordnet werden, so daß der Fremdschlüssel des Elements Personen als Attribut der Aktion einzutragen wäre.

Fig. 8F zeigt eine mögliche Struktur einer Aktionskategorietabelle tblZCatAction.

Die Aktionskategorietabelle tblZCatAction legt für jede einzelne Operation fest, ob es sich um eine Input- (Dateneingabe oder -änderung) oder Output- (Datenausgabe-)Operation handelt.

Die Tabelle tblZCatAction dient – wie soeben bei Tabelle tblZAction dargestellt – der Strukturierung von Aktionen in Eingabe- und Ausgabeaktionen. Dementsprechend weist sie lediglich zwei Datensätze auf. Identifikationsschlüssel ist das Attribut ZCatActionID (Inkrement; Feldtyp Long Integer) die Attributwerte Input und Output werden dem Attribut ZCatAction (Feldtyp Text) zugeordnet.

Fig. 8G zeigt eine mögliche Struktur einer Kategorierelationstabellle tblZCatRel.

Zwischen der Tabelle tblZRel (für Relationship/Beziehungsrichtungen) und der Tabelle tblZCat (für Categories) besteht eine n : m-Beziehung. Der Beziehungsrichtung ITM-PRS können z. B. n Beziehungskategorien zugeordnet werden (z. B. werden der Beziehung ITM-PRS die Kategorien Eigentümer, Besitzer, früherer Eigentümer etc. zugeordnet); umgekehrt können – wie oben bereits ausgeführt – einer Beziehungskategorie mehrere Beziehungstypen zugeordnet sein (so wird die Kategorie Eigentümer nicht nur im Verhältnis ITM-PRS, sondern – wie unter Eigenschaften einer Word-Datei festzustellen ist – auch im Rahmen der Dateibeschreibung verwendet (also hier in der Beziehung HYP-PRS)).

Die n : m-Beziehung wird in der Tabelle tblZCatRel aufgelöst, die einen aus den Fremdschlüssen der Tabellen tblZCat und tblZRel gebildeten, zusammengesetzten Identifikationsschlüssel – bestehend aus den Attributen ZCatID (Feldtyp Zahl, Long Integer) und ZRelID (Feldtyp Zahl, Long Integer) – besitzt. Das Attribut ZCatRelTxt (Feldtyp Text) dient zur Aufnahme einer Anmerkung, die z. B. als Hilfestellung bei der Dateneingabe dienen kann. Die Notiz ist funktional abhängig von der Kategorie-Beziehungstyp-Beziehung. Wird z. B. die Kategorie Autor in den Beziehungstypen OBJ-PRS und HYP-PRS verwendet, kann die Hilfestellung im Rahmen der Autor-OBJPRS-Beziehung z. B. lauten "Bitte wählen Sie den Autor der Publikation aus oder legen Sie ihn als Person neu an.", während die Hilfestellung im Rahmen der Autor-HYPPRS-Beziehung z. B. lautet "Bitte wählen Sie die in der Dateieigenschaft als

Autor angegebene Person aus bzw. legen Sie diese neu an."

Fig. 8H zeigt eine mögliche Struktur einer Elementbeziehungsstabelle tblZRelEle.

Die n : m-Beziehung zwischen den Tabellen tblZRel (Relationships/Beziehungsrichtungen) und tblZEle (Elemente/Klassen) wird durch die Beziehungstabelle tblZRelEle aufgelöst.

Fig. 8I zeigt eine mögliche Struktur einer Unterklassenbeziehungstabelle tblZSubEleCat.

10 Zwischen den Subelementen bzw. Subklassen (veraltet in tblSubEle) und den Beziehungskategorien (veraltet in tblCat) besteht eine n : m-Beziehung. Ein Subelement – z. B. die Stiftung – kann über beliebig viele Beziehungskategorien (z. B. Rechtsform, Stiftungszweck, Destinär etc.) definiert werden. Eine Beziehungskategorie (z. B. Rechtsform) kann jedoch umgekehrt beliebig vielen Subelementen zugeordnet werden (z. B. neben der Stiftung z. B. auch noch anderen Subelementen von Personen).

15 Die n : m-Beziehung wird in der Tabelle tblZSubEleCat aufgelöst. Der zusammengesetzte Identifikationsschlüssel besteht hier aus den Fremdschlüssen ZSubEleID (Zahl, Feldtyp Long Integer) und ZCatID (Zahl, Long Integer) der Tabelle tblZSubEle und tblZCat.

Das Attribut Standard (Feldtyp Text) dient zur Auf-

20 nahme eines Standardwertes zur Vorbereitung einer Anwendungsfunktion, die bei der Erstellung eines dynamischen Formulars z. B. zur Neuanlage einer Gesellschaft neben den für Gesellschaften vorgesehenen Informationskategorien auch entsprechende Standardwerte vorsieht. So kann z. B. für die Kategorie Rechtsform bei der Neuanlage einer Gesellschaft als Standardwert die KeyID aus der Tabelle tblKey eingetragen werden, die das Wort "Stiftung" repräsentiert.

25 In dem Attribut ZSubEleCatTxt (Feldtyp Text) können Anmerkungen und Hilfestellungen eingetragen werden.

Fig. 8J zeigt eine mögliche Struktur einer Kategorieworttabellle tblZCatKey.

Die Tabelle tblZCatKey kann optional vorgesehen werden, um den Beziehungskategorien bestimmte Auswahlwerte aus dem Bereich der Wörter zuzuordnen. So könnten z. B. der Beziehungskategorie Rechtsform Wörter wie Stiftung des bürgerlichen Rechts, GmbH, AG, eG etc. zugeordnet werden, die im Bereich der zu erstellenden Anwendung Auswahllisten ermöglichen.

30 40 Die Tabelle tblZCatKey besteht lediglich aus einem zusammengesetzten Identifikationsschlüssel. Die beiden Fremdschlüsse sind die Attribute ZCatID (Zahl, Feldtyp Long Integer; Fremdschlüssel aus Tabelle tblZCat) und Keya (Zahl, Feldtyp Long Integer; Fremdschlüssel aus Tabelle tblKey).

45 Fig. 9A, 9B und 9C zeigen schematisch die Verknüpfung der Grundtabellen tblKey, tblMmo bzw. tblHyp mit den weiteren Grundtabellen des Datenbanksystems sowie mit sich selbst über entsprechende Beziehungstabellen, wobei die jeweilige Kategorien der Beziehungen in der Kategorietabelle tblZCat definiert sind.

50 55 Fig. 10 zeigt eine schematische Darstellung der Metabene des erfahrungsgemäßen Datenbanksystems, welche die anhand der einzelnen Tabellen beschriebenen Funktionen ausführt.

60 Die Inhalte der Tabellen können über die mengenorientierte Sprache SQL (Structure Query Language) abgefragt und manipuliert werden. Die Abfragen dienen u. a. dazu, die in verschiedenen Tabellen gespeicherten, logisch zusammengehörigen Daten als einheitliche Menge wiederum in Tabellenform darzustellen. An den auf diese Weise in Abfragen ausgewählten Daten können Manipulationen, wie Lösen, Aktualisieren, Erstellen neuer Tabellen, Anfügen an

bestehende Tabellen, vorgenommen werden.

In den Diagrammen der Fig. 9A bis 9C sowie der Fig. 10 werden Elementtabellentypen durch rechteckige Kästen dargestellt, während die Beziehungstabellentypen als Rauten dargestellt werden. Zusätzlich könnten Attribute einzelner Tabellen als Ovale dargestellt werden. Da sich die Attribute bei der Erfindung jedoch auch aus dem relationalen Datenbankschema selbst ergeben können, wird auf die Darstellung von Attributen verzichtet.

#### Patentansprüche

1. Rechnergestütztes System zum Aufbauen einer Datenbank, umfassend wenigstens zwei Grundtabellen (tblKey, tblMmo) zur Speicherung von zu jeweils einer Grundklasse gehörenden Elementen, 15  
wenigstens eine Beziehungstabelle (tblKeyKey), zur Speicherung von Beziehungen zwischen den Elementen, die mit den Grundtabellen verknüpfbar ist, und eine Kategorietabelle (tblZCat) zur Speicherung von 20 Beziehungskategorien, die mit der Beziehungstabelle verknüpfbar ist, um den jeweiligen Beziehungen eine eindeutig Beziehungskategorie zuzuordnen.
2. System nach Anspruch 1, dadurch gekennzeichnet, daß mehrere Beziehungstabellen (tblKeyKey) vorgesehen sind, um jeweils zwischen den Elementen aus einer Grundtabelle oder aus zwei verschiedenen Grundtabellen Beziehungen herzustellen. 25
3. System nach einem der vorangehenden Ansprüche, dadurch gekennzeichnet, daß jede Grundtabelle Identifikationsschlüssel (KeyID) enthält, welche die Elemente in der Grundtabelle eindeutig kennzeichnen. 30
4. System nach einem der vorangehenden Ansprüche, dadurch gekennzeichnet, daß die Beziehungstabelle aus mehreren Fremdschlüsseln (KeyID, Key1ID) gebildete Beziehungsschlüssel (CatKeyKeyID) enthält, die eine Verknüpfung von jeweils zwei Elementen aus einer Grundtabelle oder aus zwei verschiedenen Grundtabellen herstellen. 35
5. System nach Anspruch 3 und 4, dadurch gekennzeichnet, daß die Fremdschlüssel auf den Identifikationsschlüsseln der zugehörigen Elemente basieren. 40
6. System nach Anspruch 4 oder 5, dadurch gekennzeichnet, daß die Beziehungsschlüssel eine Verknüpfung zu einer Beziehungskategorie in der Kategorietabelle herstellen. 45
7. System nach einem der vorangehenden Ansprüche, gekennzeichnet durch eine Relationstabelle (tblZRel), welche die möglichen Beziehungen zwischen den Grundtabellen enthält und der Kategorietabelle verknüpfbar ist, um jeweils einer Beziehungskategorie mögliche Beziehungen zwischen den Grundtabellen zuzuordnen. 50
8. System nach Anspruch 7, gekennzeichnet durch eine Kategorie-Relationstabelle (tblZCatRel), welche die Kategorietabelle (tblZCat) mit der Relationstabelle (tblZRel) verknüpft und eine Beziehung einschließlich einer Beziehungsrichtung für die Verknüpfung jeweils zweier Elemente kennzeichnet. 55
9. System nach einem der vorangehenden Ansprüche, gekennzeichnet durch wenigstens eine Untertabelle (tblZSubEle) für eine Untergruppe von zu einer Grundklasse gehörenden Elementen, die über eine Beziehungs-Untertabelle (tblZSubEleCat) mit der Kategorietabelle (tblZCat) verknüpfbar ist, um Beziehungskategorien zu bestimmen, die der Untergruppe zugeordnet werden sollen. 60
10. System nach einem der vorangehenden Ansprüche,

gekennzeichnet durch eine Aktionstabelle (tblZAction), die Eingabe/Ausgabeaktionen zur Erstellung und Nutzung der Datenbank enthält.

11. System nach einem der vorangehenden Ansprüche, dadurch gekennzeichnet, daß die Tabellen in zugeordneten Speicherbereichen realisiert sind.

12. System nach einem der vorangehenden Ansprüche, dadurch gekennzeichnet, daß die Grundklassen (tblKey, tblMmo, tblHyp, tblPrs, tblItm, tblLoc, tblPrc, tblPrj, tblObj) wenigstens zwei der folgenden Klassen umfassen: Wort, Aussage, Hyperlink, Person, Sache, Ort, Zeit, Projekt und Objekt; vorzugsweise immer die Klasse Wort.

13. Verfahren zum Aufbauen eines Datenbanksystems in einer elektronischen Datenverarbeitungsanlage, mit folgenden Verfahrensschritten:

Vorsehen wenigstens zweier Grundtabellen (tblKey, tblMmo) zur Speicherung von jeweils zu einer Grundklasse gehörenden Elementen,

Vorsehen wenigstens einer Beziehungstabelle (tblKeyKey, tblKeyMmo) zur Speicherung von Beziehungen, die einer oder mehreren Grundtabelle(n) verknüpft wird, um Beziehungen zwischen den Elementen herzustellen, und

Vorsehen einer Kategorietabelle (tblZCat) zur Speicherung von Beziehungskategorien, die mit der Beziehungstabelle verknüpft wird, um den jeweiligen Beziehungen eine eindeutig Beziehungskategorie zuzuordnen.

14. Verfahren nach Anspruch 13, dadurch gekennzeichnet, daß mehrere Beziehungstabellen (tblKeyKey, tblKeyMmo) vorgesehen werden, um jeweils zwischen den Elementen aus einer Grundtabelle (tblKey, tblMmo) oder aus zwei verschiedenen Grundtabellen (tblKey, tblMmo) Beziehungen herzustellen.

15. Verfahren zum Aufbauen einer Datenbank in einer elektronischen Datenverarbeitungsanlage, mit folgenden Verfahrensschritten:

Eingeben und Speichern von zu jeweils einer Grundklasse gehörenden Elementen in wenigstens zwei Grundtabellen (tblKey, tblMmo),

Eingeben und Speichern von Beziehungen, die zwischen den Elementen in den Grundtabellen bestehen, in wenigstens einer Beziehungstabelle (tblKeyKey, tblKeyMmo) und Verknüpfen der Beziehungstabelle (tblKeyKey, tblKeyMmo) mit der Grundtabelle (tblKey, tblMmo), und

Eingeben und Speichern von Beziehungskategorien in einer Kategorietabelle (tblZCat) und Verknüpfen der Kategorietabelle mit der Beziehungstabelle, um den jeweiligen Beziehungen eine eindeutig Beziehungskategorie zuzuordnen.

16. Verfahren nach Anspruch 15, dadurch gekennzeichnet, daß mehrere Beziehungstabellen (tblKeyKey, tblKeyMmo) vorgesehen werden, die jeweils zwischen den Elementen innerhalb einer Grundtabelle oder zwischen den Elementen aus zwei verschiedenen Grundtabellen Beziehungen herstellen.

17. Verfahren nach Anspruch 15 oder 16, dadurch gekennzeichnet, daß in der Beziehungstabelle (tblKeyKey, tblKeyMmo) aus Fremdschlüsseln (KeyID, Key1ID) gebildete Beziehungsschlüssel (CatKeyKeyID) gespeichert werden, die eine Verknüpfung von jeweils zwei Elementen innerhalb einer Grundtabelle oder von jeweils zwei Elementen aus zwei verschiedenen Grundtabellen herstellen.

18. Verfahren nach Anspruch 17, dadurch gekennzeichnet, daß die Fremdschlüssel auf der Grundlage

von Identifikationsschlüsseln gebildet werden, die zu den zugehörigen Elementen in den jeweiligen Grundtabellen gespeichert werden.

19. Verfahren nach Anspruch 17 oder 18, dadurch gekennzeichnet, daß die Beziehungsschlüssel (CatKey-KeyID) eine Verknüpfung zu einer Beziehungskategorie in der Kategorietabelle (tblZCat) herstellen. 5

20. Verfahren nach einem der Ansprüche 15 bis 19, dadurch gekennzeichnet, daß eine Relationstabelle (tblZRel) vorgesehen wird, welche die möglichen Beziehungen zwischen den Grundtabellen (tblKey, tblMmo) enthält und mit der Kategorietabelle (tblZCat) verknüpft wird, um jeweils einer Beziehungskategorie mögliche Beziehungen zwischen den Grundtabellen zuzuordnen. 10

21. Verfahren nach Anspruch 20, dadurch gekennzeichnet, daß eine Kategorie-Relationstabelle (tblZCatRel) vorgesehen wird, welche die Kategorietabelle (tblZCat) mit der Relationstabelle (tblZRel) verknüpft, um eine Beziehung einschließlich einer Beziehungsrichtung für die Verknüpfung jeweils zweier Elemente zu kennzeichnen. 15

22. Verfahren nach einem der Ansprüche 13 bis 21, dadurch gekennzeichnet, daß die Grundklassen (tblKey, tblMmo, tblHyp, tblPrs, tblIIm, tblLoc, tblPrc, 25  
tblPrj, tblObj) aus wenigstens zwei der folgenden Klassen ausgewählt werden: Wort, Aussage, Hyperlink, Person, Sache, Ort, Zcit, Projekt und Objekt, wobei eine Grundklasse vorzugsweise immer die Wort-Klasse ist. 30

---

Hierzu 12 Seite(n) Zeichnungen

---

35

40

45

50

55

60

65

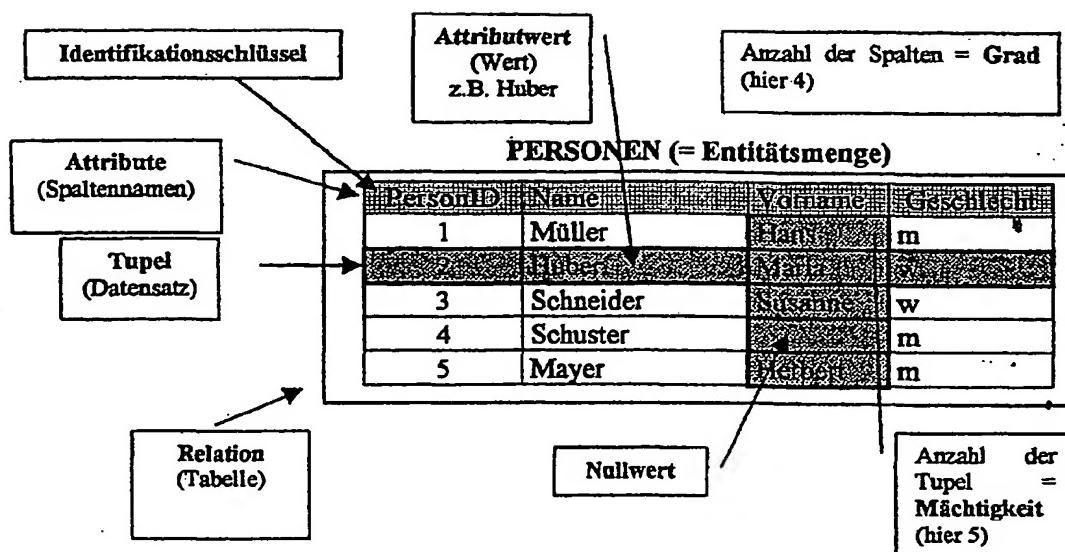


Fig. 1

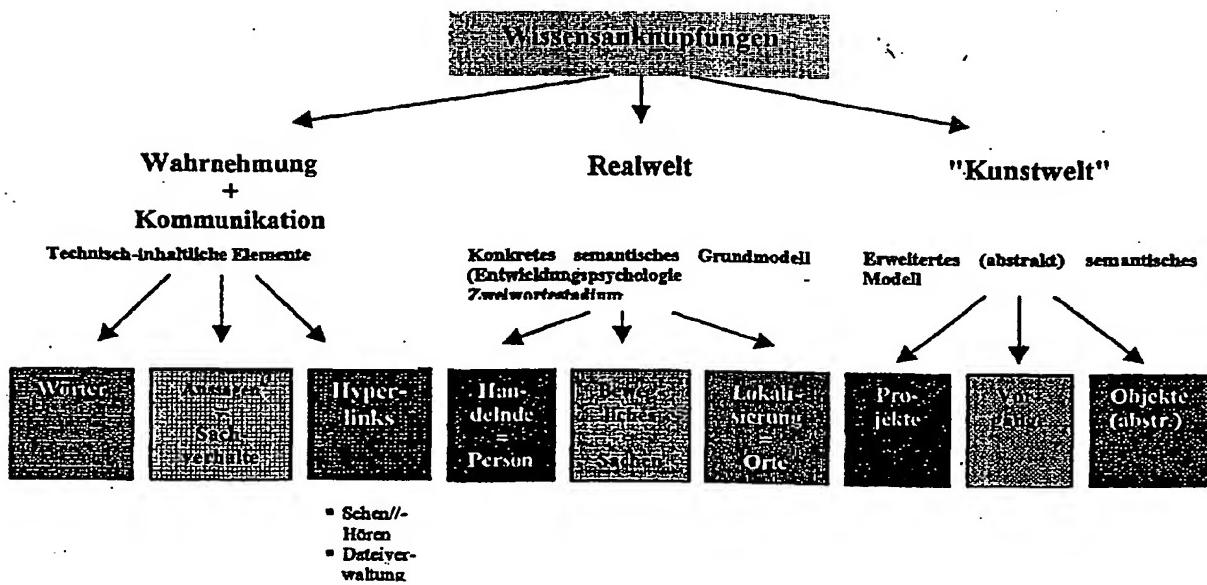


Fig. 2

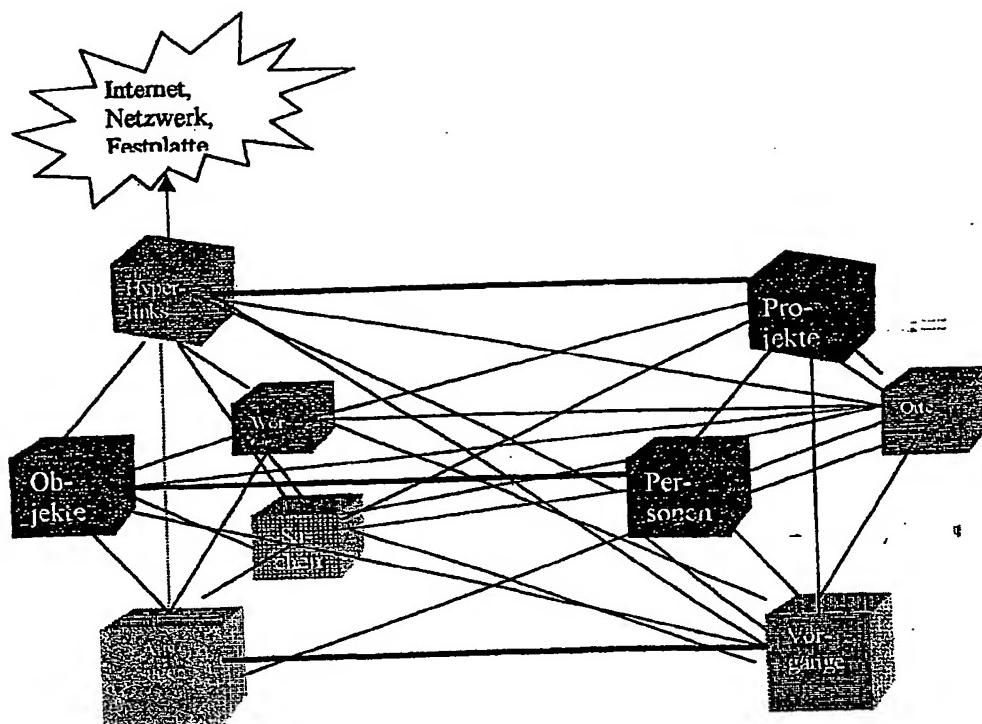


Fig. 3

	Wörter	Aussagen	Hyperlinks	Personen	Sachen	Orte	Projekte	Vorgänge	Objekte
Wörter	Wort-Aussage Wort-Wort	Aussage-Wort	Hyperlinks Wort	Person-Wort	Sache-Wort	Ort-Wort	Projekt-Wort	Vorgang-Wort	Objekt-Wort
Aussagen	Wort-Aussage Aussage-Aussage	Aussage-Aussage	Hyperlinks Aussage	Person-Aussage	Sache-Aussage	Ort-Aussage	Projekt-Aussage	Vorgang-Aussage	Objekt-Aussage
Sehen/Hören	Wort-Hörer Hörer-Hörer	Aussage-Hörer	Hyperlinks Hörer	Person-Hörer	Sache-Hörer	Ort-Hörer	Projekt-Hörer	Vorgang-Hörer	Objekt-Hörer
Personen	Wort-Person Person	Aussage-Person	Hyperlinks Person	Person-Person	Sache-Person	Ort-Person	Projekt-Person	Vorgang-Person	Objekt-Person
Sachen	Wort-Sache Sache	Aussage-Sache	Hyperlinks Sache	Person-Sache	Sache-Sache	Ort-Sache	Projekt-Sache	Vorgang-Sache	Objekt-Sache
Orte	Wort-Ort Ort	Aussage-Ort	Hyperlinks Ort	Person-Ort	Sache-Ort	Ort-Ort	Projekt-Ort	Vorgang-Ort	Objekt-Ort
Projekte	Wort-Projekt Projekt	Aussage-Projekt	Hyperlinks Projekt	Person-Projekt	Sache-Projekt	Ort-Projekt	Projekt-Projekt	Vorgang-Projekt	Objekt-Projekt
Vorgänge	Vorgang-Vorgang	Aussage-Vorgang	Hyperlinks Vorgang	Person-Vorgang	Sache-Vorgang	Ort-Vorgang	Projekt-Vorgang	Vorgang-Vorgang	Objekt-Vorgang
Objekte	Wort-Objekt Objekt	Aussage-Objekt	Hyperlinks Objekt	Person-Objekt	Sache-Objekt	Ort-Objekt	Projekt-Objekt	Vorgang-Objekt	Objekt-Objekt

Fig. 4

	Wörter	Aussagen	Hyperlinks	Personen	Sachen	Orte	Projekte	Vorgänge	Objekte
Wörter	Wörter Wörter Wörter Wörter	Aussage-Wort Aussage-Wort Aussage-Wort Aussage-Wort	Hyperlink-Wort Hyperlink-Wort Hyperlink-Wort Hyperlink-Wort	Person-Wort Person-Wort Person-Wort Person-Wort	Sache-Wort Sache-Wort Sache-Wort Sache-Wort	Ort-Wort Ort-Wort Ort-Wort Ort-Wort	Projekt-Wort Projekt-Wort Projekt-Wort Projekt-Wort	Vorgang-Wort Vorgang-Wort Vorgang-Wort Vorgang-Wort	Objekt-Wort Objekt-Wort Objekt-Wort Objekt-Wort
Aussagen	Wort-Aussage Aussage-Aussage	Aussage-Aussage Aussage-Aussage	Hyperlink-Aussage Hyperlink-Aussage	Person-Aussage Person-Aussage	Sache-Aussage Sache-Aussage	Ort-Aussage Ort-Aussage	Projekt-Aussage Projekt-Aussage	Vorgang-Aussage Vorgang-Aussage	Objekt-Aussage Objekt-Aussage
Sachen/Hör-en	Wörter-Hyperlink Hyperlink-Hyperlink	Aussage-Hyperlink Hyperlink-Hyperlink	Hyperlink-Hyperlink Hyperlink-Hyperlink	Person-Hyperlink Person-Hyperlink	Sache-Hyperlink Sache-Hyperlink	Ort-Hyperlink Hyperlink	Projekt-Hyperlink Hyperlink	Vorgang-Hyperlink Hyperlink	Objekt-Hyperlink Hyperlink
Personen	Wort-Person Person-Person	Aussage-Person Person-Person	Hyperlink-Person Hyperlink-Person	Person-Person Person-Person	Sache-Person Person-Person	Ort-Person Person-Person	Projekt-Person Person-Person	Vorname-Person Person	Objekt-Person Person
Sachen	Wort-Sache Sache-Sache	Aussage-Sache Sache-Sache	Hyperlink-Sache Sache-Sache	Person-Sache Person-Sache	Sache-Sache Sache-Sache	Ort-Sache Sache-Sache	Projekt-Sache Sache-Sache	Vorgang-Sache Sache-Sache	Objekt-Sache Sache-Sache
Orte	Wort-Ort Ort-Ort	Aussage-Ort Ort-Ort	Hyperlink-Ort Ort-Ort	Person-Ort Person-Ort	Sache-Ort Ort-Ort	Ort-Projekt Ort-Ort	Projekt-Ort Ort-Ort	Vorgang-Ort Ort-Ort	Objekt-Ort Ort-Ort
Projekte	Wort-Projekt Projekt-Projekt	Aussage-Projekt Projekt-Projekt	Hyperlink-Projekt Projekt-Projekt	Person-Projekt Projekt-Projekt	Sache-Projekt Projekt-Projekt	Ort-Projekt Projekt-Projekt	Projekt-Projekt Projekt-Projekt	Vorname-Projekt Projekt-Projekt	Objekt-Projekt Projekt-Projekt
Vorgänge	Wort-Vorgang Vorgang-Vorgang	Aussage-Vorgang Vorgang-Vorgang	Hyperlink-Vorgang Vorgang-Vorgang	Person-Vorgang Vorgang-Vorgang	Sache-Vorgang Vorgang-Vorgang	Ort-Vorgang Vorgang-Vorgang	Projekt-Vorgang Vorgang-Vorgang	Vorname-Vorgang Vorgang-Vorgang	Objekt-Vorgang Vorgang-Vorgang
Objekte	Wort-Objekt Objekt-Objekt	Aussage-Objekt Objekt-Objekt	Hyperlink-Objekt Objekt-Objekt	Person-Objekt Objekt-Objekt	Sache-Objekt Objekt-Objekt	Ort-Objekt Objekt-Objekt	Projekt-Objekt Objekt-Objekt	Vorname-Objekt Objekt-Objekt	Objekt-Objekt Objekt-Objekt

Fig. 4a

**TBLKEY**

<b>KeyID</b>	<b>KeyName</b>	<b>KeyRank</b>

Fig. 5a

**TBLMMO**

<b>MmoID</b>	<b>Mmo</b>	<b>MmoCreated</b>	<b>MmoChanged</b>

Fig. 5b

**TBLHYP**

<b>HypID</b>	<b>Hyperlink</b>	<b>HypDCC</b>

Fig. 5c

**TBLPRS**

<b>PrsID</b>	<b>PrsCreated</b>	<b>PrsChanged</b>	<b>PrsDCC</b>

Fig. 5d

**TBLITM**

<b>ItmID</b>	<b>ItmCreated</b>	<b>ItmChanged</b>	<b>ItmDCC</b>

Fig. 5e

**TBLLOC**

<b>LocID</b>	<b>LocCreated</b>	<b>LocChanged</b>	<b>LocDCC</b>

Fig. 5f

**TBLPRJ**

<b>PrjID</b>	<b>PrjCreated</b>	<b>PrjChanged</b>	<b>PrjDCC</b>

Fig. 5g

**TBLOBJ**

<b>ObjID</b>	<b>ObjCreated</b>	<b>ObjChanged</b>	<b>ObjDCC</b>

Fig. 5h

**TBLPRC**

<b>PrcID</b>	<b>PrcStart</b>	<b>PrcEnd</b>	<b>PrcDCC</b>

Fig. 5i

**TBLCATPRSPRSTXT**

<b>CatPrsPrsTxtID</b>	<b>CatPrsPrsTxt</b>

Fig. 5j

Fig. 6a

**TBLKEYKEY**

<b>KeyID</b>	<b>Key1ID</b>	<b>CatKeyKeyID</b>	<b>KeyKey123</b>	<b>KeyKeyTxt</b>	<b>KeyKeyActivate</b>

Fig. 6b

**TBLKEYMMO**

<b>KeyID</b>	<b>MmoID</b>	<b>CatKeyMmoID</b>	<b>KeyMmo123</b>	<b>KeyMmoTxt</b>	<b>KeyMmoActivate</b>

Fig. 6c

**TBLKEYHYP**

<b>KeyID</b>	<b>HypID</b>	<b>CatKeyHypID</b>	<b>KeyHyp123</b>	<b>KeyHypTxt</b>	<b>KeyHypActivate</b>

Fig. 6d

**TBLMMOMOMMO**

<b>MmoID</b>	<b>Mmo1ID</b>	<b>CatMmoMmoID</b>	<b>MmoMmo123</b>	<b>MmoMmoTxt</b>	<b>MmoMmoActivate</b>

Fig. 6e

**TBLMMOHYP**

<b>MmoID</b>	<b>HypID</b>	<b>CatMmoHypID</b>	<b>MmoHyp123</b>	<b>MmoHypTxt</b>	<b>MmoHypActivate</b>

Fig. 6f

**HYPHYP**

<b>HypID</b>	<b>Hyp1ID</b>	<b>CatHypHypID</b>	<b>HypHyp123</b>	<b>HypHypTxt</b>	<b>HypHypActivate</b>

Fig. 7

**TBLZCAT**

ZCatID	ZCat	ZCatRnk

Fig. 8a

**TBLZREL**

ZRelID	ZRel	ZEle_isID	ZRelT2	ZRelRnk

Fig. 8b

**TBLZELE IS**

ZEle_isID	ZEle_is

Fig. 8c

**TBLZELE**

ZEleID	ZEle	ZEleRnk

Fig. 8d

**TBLZSUBELE**

ZSubEleID	ZSubEle	ZSubEleRnk	ZEleID

Fig. 8e

**TBLZACTION**

ZActionID	ZAction	ZCatActionID	ZEleID

Fig. 8 f

**TBLZCATACTION**

ZCatActionID	ZCatAction
1	Input
2	Output

Fig. 8 g

**TBLZCATREL**

ZCatID	ZRelID	ZCatRelTxt

Fig. 8 h

**TBLZRELELE**

ZRelID	ZEleID

Fig. 8 i

**TBLZSUBELECAT**

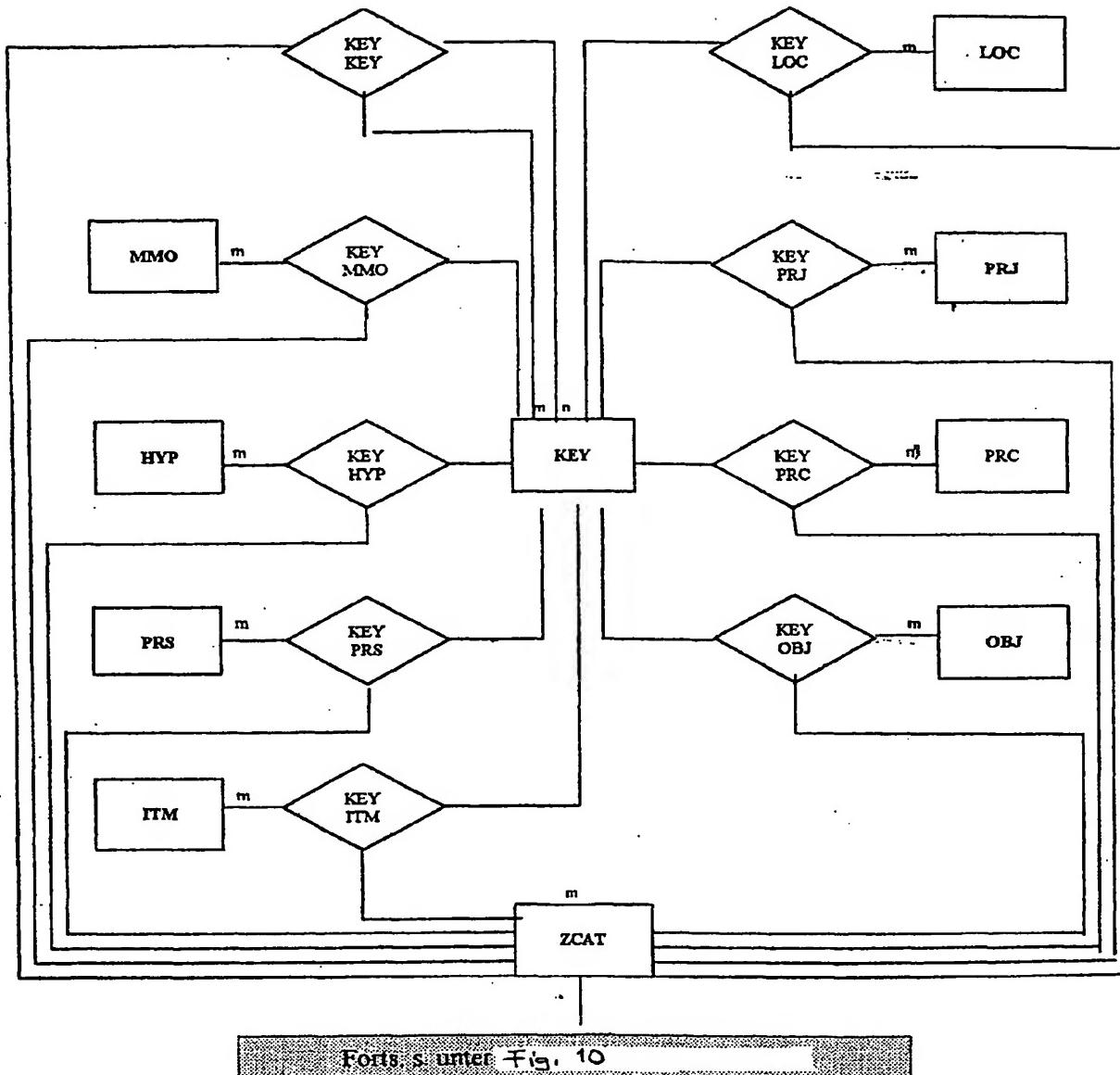
ZSubEleID	ZCatID	Standard	ZSubEleCatTxt

Fig. 8 j

**TBLZCATKEY**

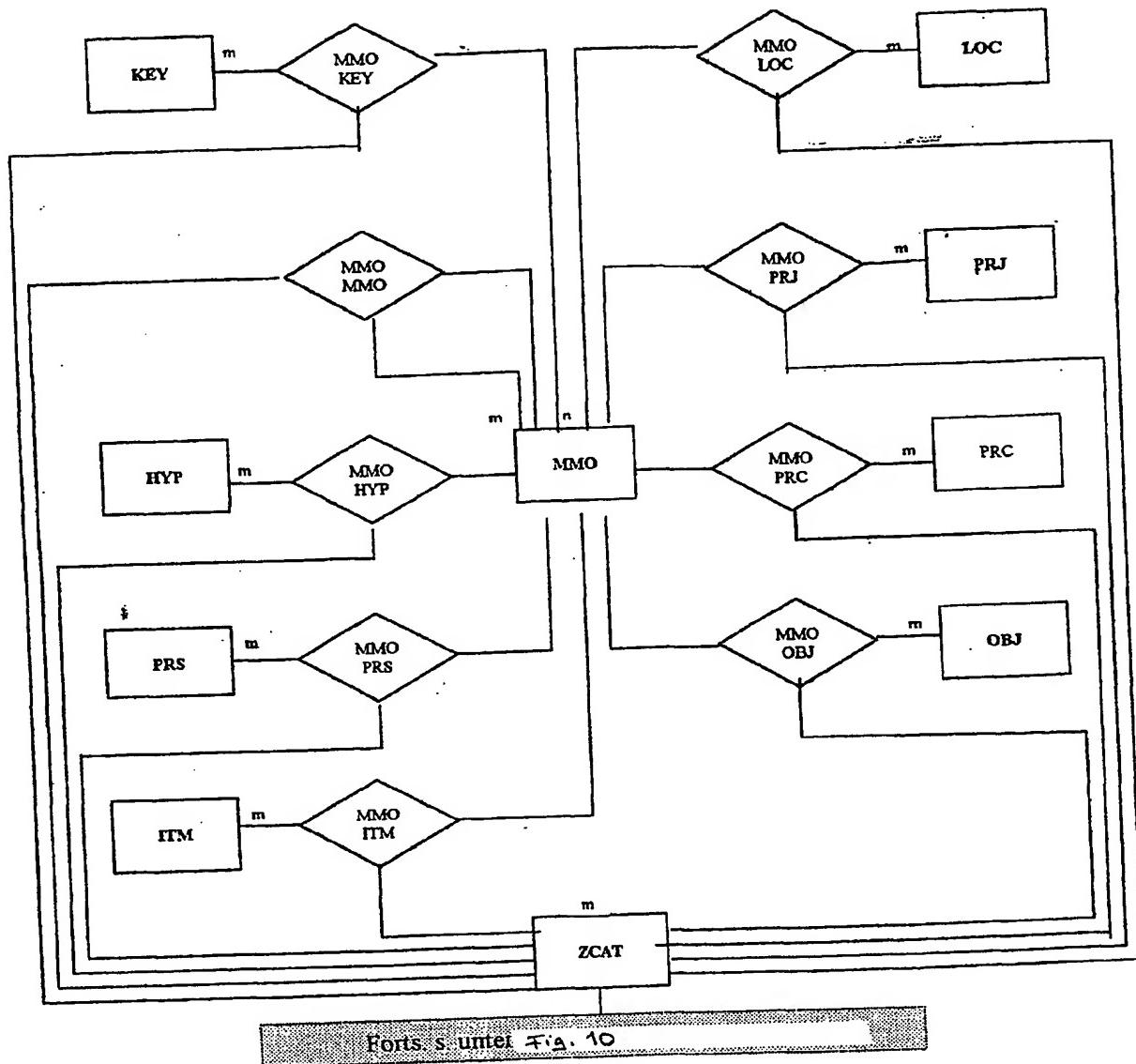
ZCatID	KeyID

Fig. 9a



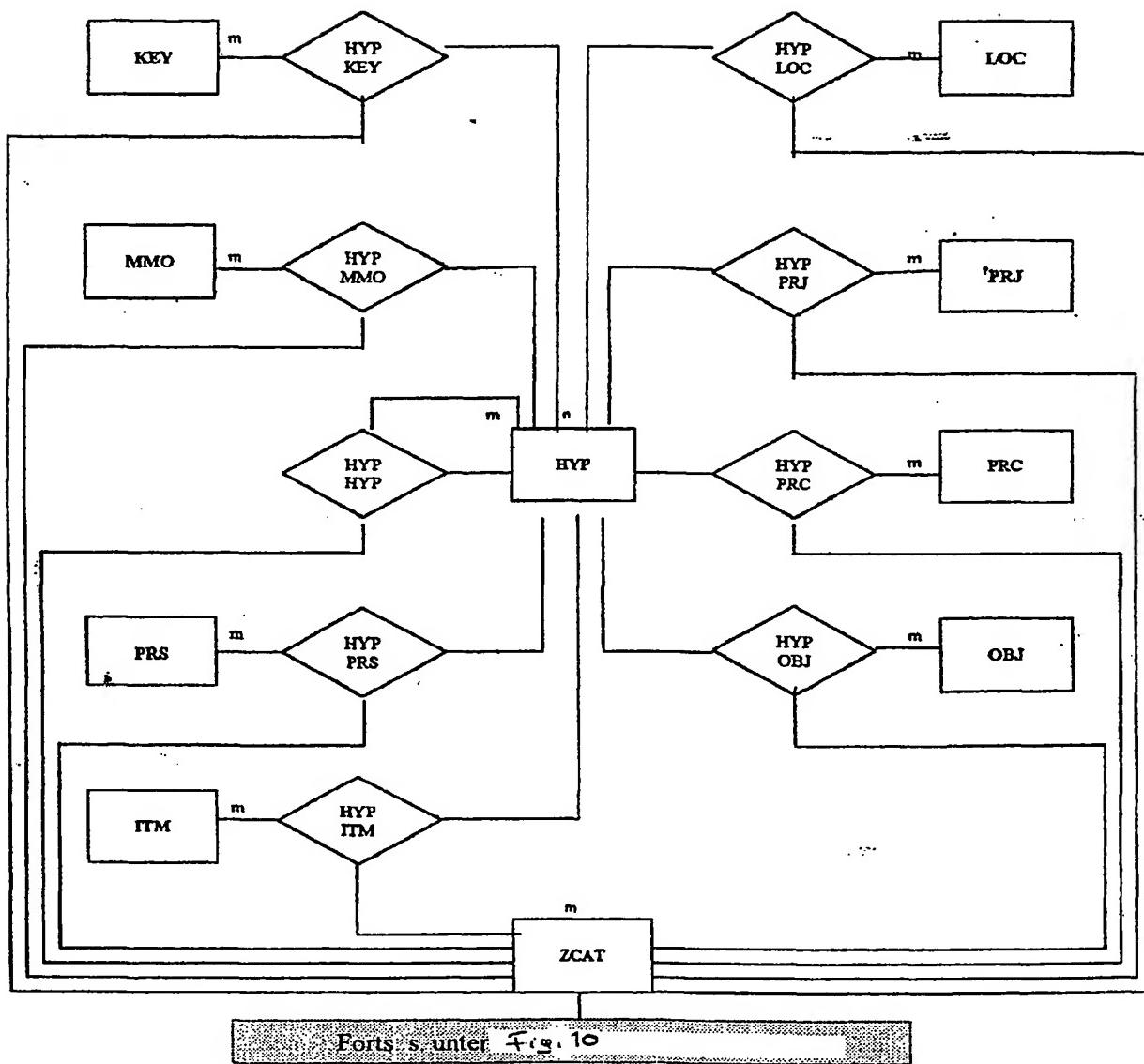
Forts. s. unter Fig. 10

Fig. 9b



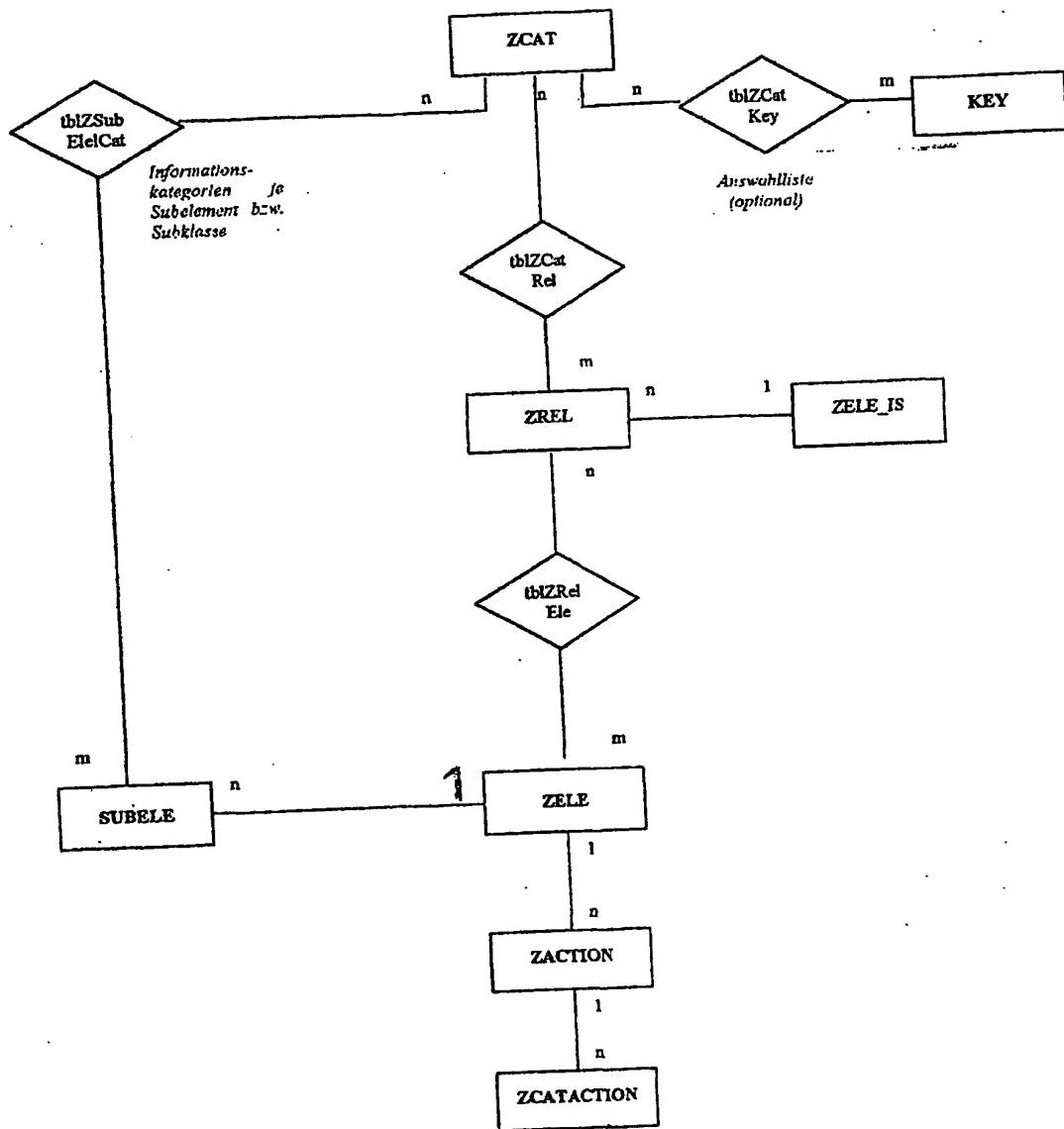
Forts. s. unter Fig. 10

Fig. 9c



Forts. unter Fig. 10

Fig. 10



**Computer based system for construction of a relational database with a procedure for inputting data with an extra relationship category table so that each relationship is assigned to a unique relationship category**

**Patent number:** DE19914454  
**Publication date:** 2000-10-05  
**Inventor:** DINGLREITER MARCUS (DE)  
**Applicant:** DINGLREITER MARCUS (DE)  
**Classification:**  
- **international:** G06F17/30  
- **european:**  
**Application number:** DE19991014454 19990330  
**Priority number(s):** DE19991014454 19990330

**Abstract of DE19914454**

System has two base tables for storage of elements belonging to a base class, a relational table for storage of relationships between elements that is linked to the base tables and a category table for storage of relationship categories that is linked to the relationship table so that each relationship is assigned to only one relationship category. An Independent claim is made for a procedure for use with the above system.

---

Data supplied from the **esp@cenet** database - Worldwide

